



KONGU ENGINEERING COLLEGE
(Autonomous)
PERUNDURAI, ERODE - 638 060

DEPARTMENT OF INFORMATION TECHNOLOGY

LABORATORY MANUAL

22ITL61 - CLOUD COMPUTING LABORATORY



22ITL61 - CLOUD COMPUTING LABORATORY																								
Programme & Branch	B.Tech. & Information Technology						Sem.	Category	L	T	P	Credit												
Prerequisites	Operating Systems, Computer Networks						6	PC	0	0	2	1												
Preamble	This course enables the students to design, develop, and deploy cloud-based web applications.																							
LIST OF EXPERIMENTS / EXERCISES:																								
1.	Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.																							
2.	Install a C compiler in the virtual machine created using virtual box and execute Simple Programs																							
3.	Install Google App Engine. Create hello world app and other web applications using python/java																							
4.	Use GAE launcher to launch web applications																							
5.	Create EC2-AWS S3 bucket based static web pages																							
6.	Create EC2-AWS- instance and migration																							
7.	Create EC2-AWS web application using Beanstalk																							
8.	Perform AWS load balancing and auto scaling																							
9.	Implement PaaS-Mobile sensor based IoT application hosted via PaaS environment																							
10.	Install Hadoop single node cluster and run simple applications like wordcount.																							
Total:30																								
REFERENCES/ MANUAL /SOFTWARE:																								
1.	VMware, Google App Engine																							
2.	C/Python/Java																							
3.	Hadoop																							
COURSE OUTCOMES: On completion of the course, the students will be able to										BT Mapped (Highest Level)														
CO1	experiment with various virtualization tools such as Virtual Box and VMware workstation.										Applying (K3), Precision (S3)													
CO2	develop EC2-AWS buckets, instances and web applications										Applying (K3), Precision (S3)													
CO3	apply large data sets in a parallel environment.										Applying (K3), Precision (S3)													
Mapping of Cos with POs and PSOs																								
COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2										
CO1	3	2	1	1	3	3	3	3	2	2	3	2	3	2										
CO2	3	2	1	1	3	3	3	3	2	2	3	2	3	2										
CO3	3	2	1	1	3	3	3	3	2	2	3	2	3	2										

1 – Slight, 2 – Moderate, 3 – Substantial, BT- Bloom's Taxonomy

Rubrics for Assessment

Criteria/Marks assigned	Need to improve	Satisfactory	Good
Conduct of experiment (25)			
Ability to analyze problem and identify requirements (10)	Able to analyze the partial problem (5-1)	Able to analyze the problem and identify partial input and output (9-6)	Able to analyze the problem and identify correctly all input and output (10)
Demonstrate understanding on configuration & design (15)	Able to explain some Configuration & design (9-1)	Able to explain Configuration & design (14-10)	Able to explain Configuration & design Correctly and provide alternative solutions (15)
Output & Record (25)			
Ability to run/debug (15)	Able to run but have logic Errors (9-1)	Able to run correctly without any logic error and display inappropriate output (14-10)	Able to run correctly without any logic error and display appropriate output (15)
Adherence experiment completion (10)	Delayed completion (5-1)	Completion on next day (9-6)	On time completion of the experiment (10)
Viva (10)			
Able to recall and demonstrate a insightful understanding of the problem domain (10)	Minimal recall and demonstrates a minimal insightful understanding of the problem domain. (4-1)	Can recall some and demonstrates some understanding of the problem domain (7-5)	Able to recall and demonstrates a comprehensive and insightful understanding of the problem domain (10-8)

INDEX

S.NO	TOPIC	CO MAPPING	PAGE NO.
1.	Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.	CO1	4
2.	Install a C compiler in the virtual machine created using virtual box and execute Simple Programs	CO1	10
3.	Install Google App Engine. Create hello world app and other web applications using python/java	CO2	12
4.	Use GAE launcher to launch web applications	CO2	17
5.	Create EC2-AWS S3 bucket based static web pages	CO2	20
6.	Create EC2-AWS- instance and migration	CO2	24
7.	Create EC2-AWS web application using Beanstalk	CO2	28
8.	Perform AWS load balancing and auto scaling	CO2	35
9.	Implement PaaS-Mobile sensor based IoT application hosted via PaaS environment	CO3	41
10.	Install Hadoop single node cluster and run simple applications like wordcount	CO3	48

Ex:01	INSTALL VIRTUALBOX/VMWARE WORKSTATION WITH DIFFERENT FLAVOURS OF LINUX OR WINDOWS OS ON TOP OF WINDOWS7 OR 8
Date:	

AIM:

To install virtualbox/vmware workstation with different flavours of linux or windows os on top of windows7 or 8

PROCEDURE

Download oracle virtual box.org/download

1: Click Next on the first screen. This action tells the Wizard that you want to install VirtualBox.

2: Browse and select the location you want to install VirtualBox in. The default location is fine, but feel free to change it if you prefer. Click Next when you're ready to continue.

3: Check or uncheck the items you wish to include or exclude and click Next

Step 4: Finally, you'll see a screen asking you to confirm the installation. Click Install to install VirtualBox on Windows 10.

5 :Install the Files and packages:

Step 6: Install the Certificates:

7: Click Finish to close the Wizard after the installation and start using

VirtualBox.

8: When you will open virtualbox it will look like as shown below:

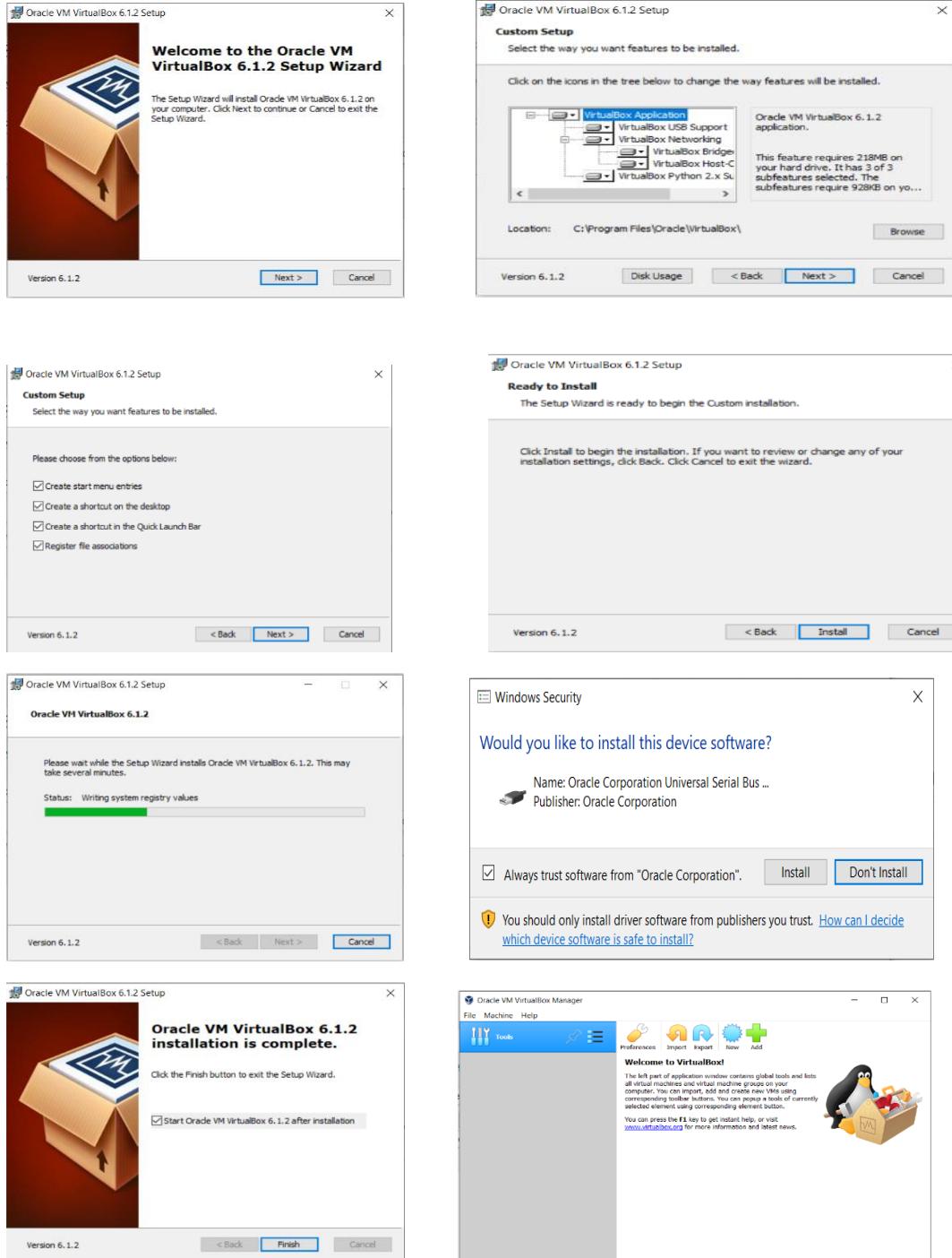
Download ios file for ubuntu

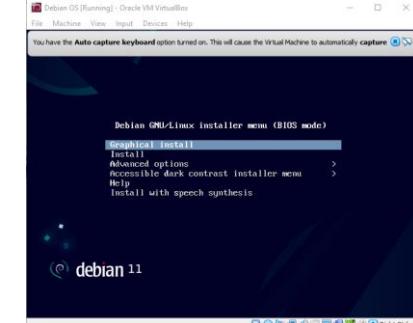
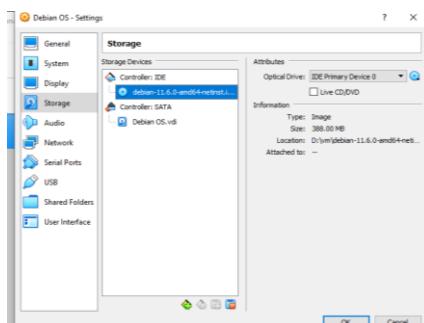
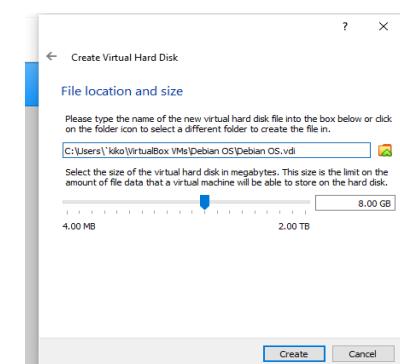
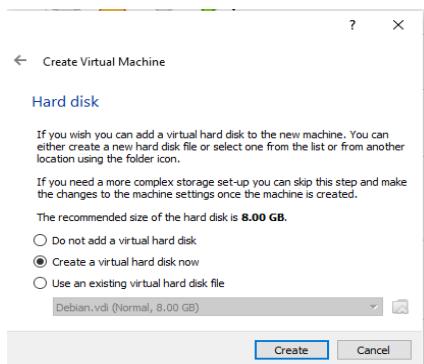
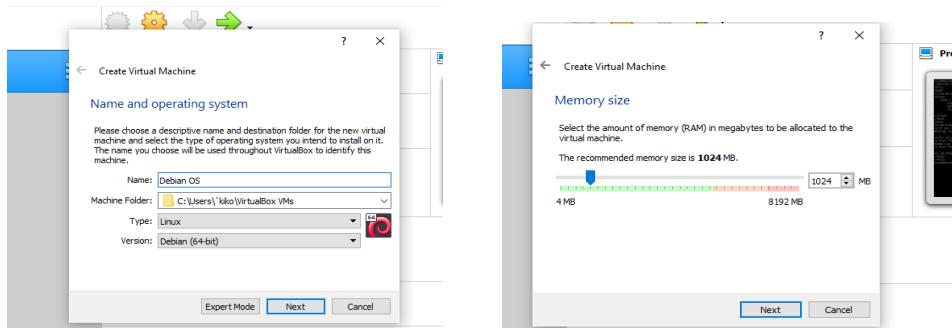
1. First, click the New button in the top-right corner of the VirtualBox window. Click the New button

2. Configure the following on the Name and operating system page.Click Next to continue.Configure the name and OS

- 3 Click Next to continue. Configure the memory size.
4. On the Hard disk screen, decide whether to create a new hard disk for your guest VM or use an existing one. This example chooses to create a new hard disk since.
5. On the Hard disk file type page, select VDI (VirtualBox Disk Image) and click Next to continue.
6. chooses the Dynamically allocated option. Click Next to continue.
7. Specify the location and maximum size of the new virtual disk, and click Create.
8. Now, select Debian GNU/Linux Live (kernel 4.19.0-5-amd64) and press Enter
9. Once it's booted, click on Activities from the top left corner and click on the Debian installer icon.
10. Select basic settings like language ,Keyboard support,location.
11. Setup users and passwords . Start by creating a root password.
12. To partition disks for Debian 10,choose the method guided-use entire disk.
13. Select the which you want to partition.
14. Confirm the changes to disks by selecting yes and select continue.
15. The installation will start.
16. To configure the package manager, your system needs to be connected to internet.
If it is connected select Yes and click continue to use the network mirror.
17. Select No to keep that information private.
18. Finished setting up and installing Debian 10 select continue to reboot the system.

OUTPUT





Set up users and passwords

You need to set a password for 'root', the system administrative account. A malicious or unqualified user with root access can have disastrous results, so you should take care to choose a root password that is not easy to guess. It should not be a word found in dictionaries, or a word that could be easily associated with you.

A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.

The root user should not have an empty password. If you leave this empty, the root account will be disabled and the system's initial user account will be given the power to become root using the "sudo" command.

Note that you will not be able to see the password as you type it.

Root password:
•••

Show Password in Clear

Please enter the same root password again to verify that you have typed it correctly.

Re-enter password to verify:
•••

Show Password in Clear

Screenshot **Go Back** **Continue**

Partition disks

The installer can guide you through partitioning a disk (using different standard schemes) or, if you prefer, you can do it manually. With guided partitioning you will still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.

Partitioning method:

- Guided - use entire disk**
- Guided - use entire disk and set up LVM
- Guided - use entire disk and set up encrypted LVM
- Manual

Screenshot **Go Back** **Continue**

Partition disks

Selected for partitioning:
SCSI (0,0,0) (sda) - ATA VBOX HARDDISK: 8.6 GB

The disk can be partitioned using one of several different schemes. If you are unsure, choose the first one.

Partitioning scheme:

- All files in one partition (recommended for new users)
- Separate /home partition
- Separate /home, /var, and /tmp partitions**

Screenshot **Go Back** **Continue**

Partition disks

This is an overview of your currently configured partitions and mount points. Select a partition to modify its settings (file system, mount point, etc.), a free space to create partitions, or a device to initialize its partition table.

Guided partitioning

- Configure software RAID
- Configure the Logical Volume Manager
- Configure encrypted volumes
- Configure iSCSI volumes

▼ SCSI (0,0,0) (sda) - 8.6 GB ATA VBOX HARDDISK

> #1 primary	2.1 GB	f	ext4	/
> #5 logical	1.0 GB	f	ext4	/var
> #6 logical	1.0 GB	f	swap	swap
> #7 logical	257.9 MB	f	ext4	/tmp
> #8 logical	4.2 GB	f	ext4	/home

Undo changes to partitions

Finish partitioning and write changes to disk

Screenshot **Help** **Go Back** **Continue**

Configure the package manager

Please select a Debian archive mirror. You should use a mirror in your country or region if you do not know which mirror has the best Internet connection to you.

Usually, deb.debian.org is a good choice.

Debian archive mirror:

- ftp.us.debian.org
- debian.csail.mit.edu
- debian.osus.org
- debian.cc.lehigh.edu
- debian.gtsc.gatech.edu
- mirror.cc.columbia.edu
- deb.debian.org**
- debian-archive.trafficmanager.net
- mirrors.hug.mtu.edu
- mirrors.us.oneandone.net
- mirrors.bloomu.edu
- mirrors.namecheap.com
- mirrors.ocf.berkeley.edu
- debian.mirror.constant.com

Screenshot **Go Back** **Continue**

Software selection

At the moment, only the core of the system is installed. To tune the system to your needs, you can choose to install one or more of the following predefined collections of software.

Choose software to install:

- Debian desktop environment
 - ... GNOME
 - ... Xfce
 - ... KDE
 - ... Cinnamon
 - ... MATE
 - ... LXDE
 - web server
 - print server
 - SSH server
 - standard system utilities

Screenshot **Go Back** **Continue**



RESULT:

Thus installation of virtualbox/vmware workstation with different flavours of linux windows or OS on top of windows7 or 8 was installed successfully.

Ex:02	INSTALL C COMPLIER IN VIRTUAL MACHINE CREATED USING VIRTUAL BOX AND EXECUTE SIMPLE PROGRAMS
Date:	

AIM:

To install c complier in virtual machine created using virtual box and execute simple programs

PROGRAMS

- 1.Start the virtual machine and log in to the operating system.
- 2.Check if the operating system is up to date by running the appropriate package manager command, such as apt-get update for Ubuntu.
- 3.Install the GCC compiler by running the appropriate package manager command, such as

apt-get install gcc for Ubuntu

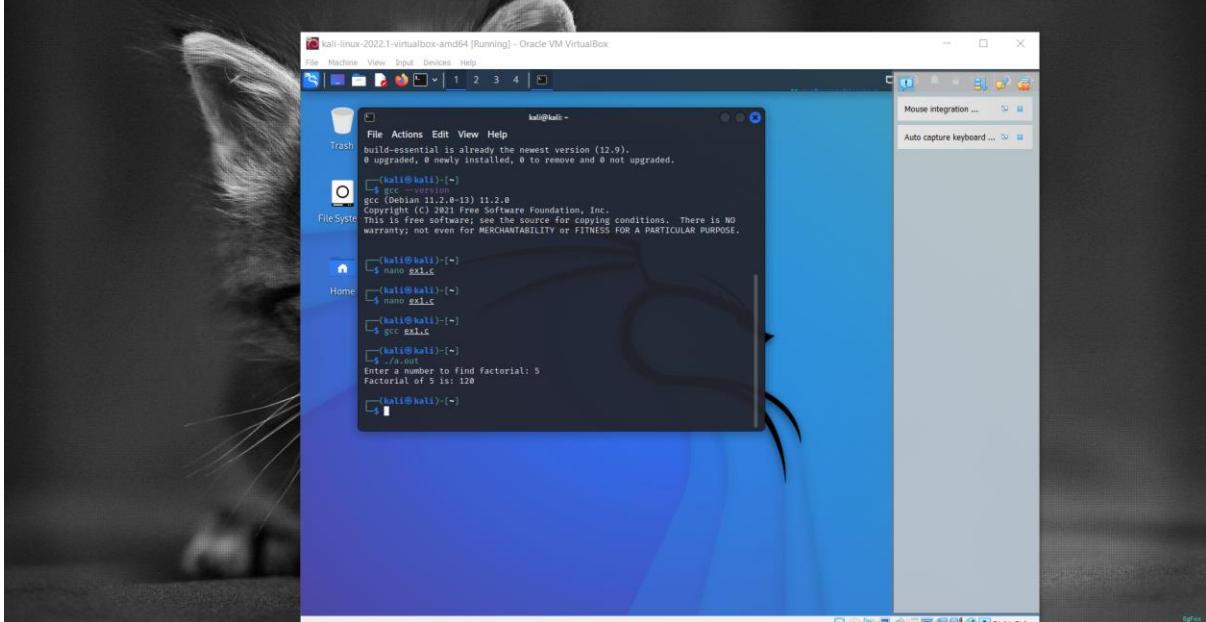
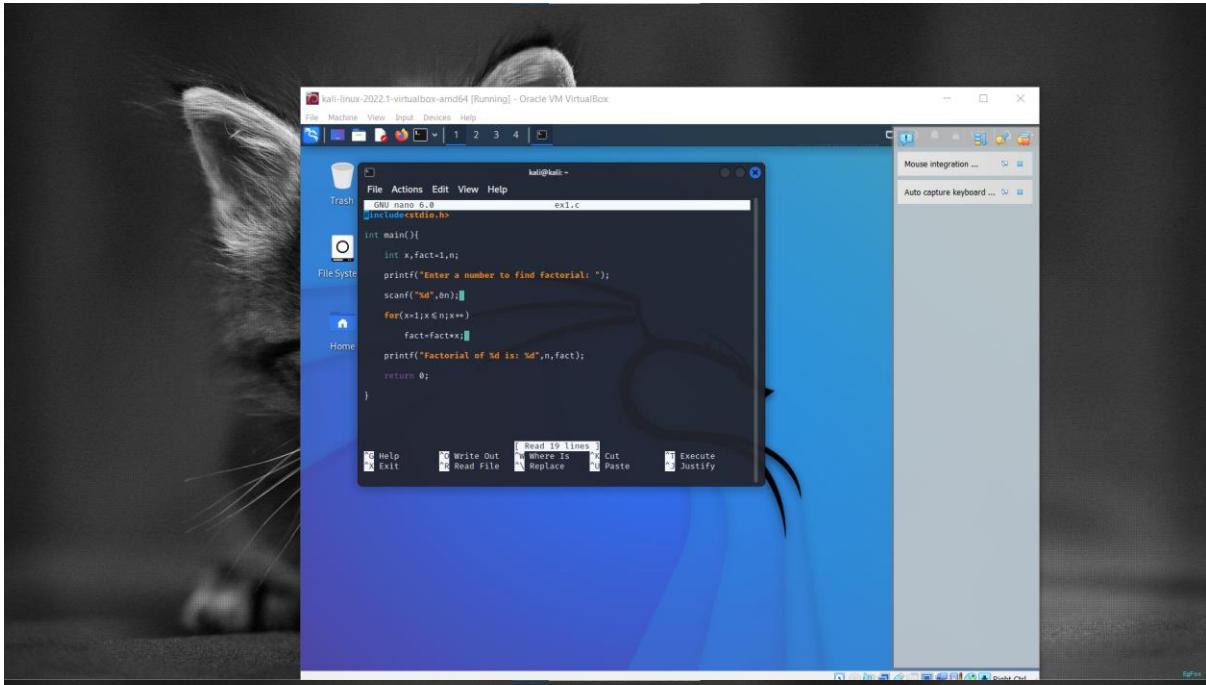
4. Verify the installation by running the gcc command in the terminal. If the installation was successful, the GCC compiler should display its usage information.
- 5.Create a simple C program, , using a text editor:
- 6.Save the program with a .c extension, such as nano filename.c
- 7.Compile the program by running the following command in the terminal:

gcc filename.c

- 8.Execute the program by running the following command in the terminal: ./a.out

The program should output on the terminal.

OUTPUT



RESULT

Thus installation of c complier in virtual machine created using virtual box and execution of simple programs was done successfully.

Ex:03	INSTALL GOOGLEAPP ENGINE CREATE HELLO WORLD APP AND WEB APP USING PYTHON
Date:	

AIM

To install google app engine create hello world app and web app using python

PROCEDURE

Step1:Download the Google Cloud SDK shell and python should be installed

Step2:Create two files named index.py and app.yaml

In index.py

```
Print("HELLO WORLD")
```

In app.yaml

```
runtime: python27
```

```
api_version: 1
```

```
threadsafe: false
```

handlers

```
- url: /
```

```
script: index.py
```

Step 3: Open google sdk shell and run as administrator and copy and paste the path of your file

```
google-cloud-sdk\bin\dev_appserver.py "path"
```

Step 4: It generate the link and Run the following link in browser

<http://localhost:8080>

OUTPUT

The screenshot shows the Google Cloud Platform homepage. At the top, there's a banner with the text "Get Started with Google Cloud Platform" and "Today \$0 free trial to get you started". Below the banner, there are sections for "Top products" featuring services like Compute Engine, Cloud Storage, Cloud SQL, and Cloud Run. On the left, a sidebar lists various services: Overview, Recent, View all products, BigQuery, IAM and admin, Marketplace, Compute Engine, Kubernetes Engine, Cloud Storage, BigQuery, VPC network, and CloudRun.

Authorise Cloud Shell

Cloud Shell needs permission to use your credentials for the gcloud command.

Click Authorize to grant permission to this and future calls.

[REJECT](#) [AUTHORISE](#)



Sign in to the gcloud CLI

You are seeing this page because you ran the following command in the gcloud CLI from this or another machine. If this is not the case, close this tab.

```
gcloud auth login --no-launch-browser
```

Enter the following verification code in gcloud CLI on the machine you want to log into. This is a credential **similar to your password** and should not be shared with others.

```
4/0AWtgh4k0GXmIIq2MbHztsoCsdz6rprnhrk  
RvpymymjrAnB5rrf1UPb-OfRj_Ei3tbzG42g
```

[Copy](#)

```
Enter authorization code: 4/0AWtgzh4k0GxmIIq2MbHztsocSdz6prnhrKRVpmymjrAnB5rrflUPb-OfRj_Ei3tbzG42g

You are now logged in as [harini8t@gmail.com].
Your current project is [None]. You can change this setting by running:
  $ gcloud config set project PROJECT_ID
harini8t@cloudshell:~$ gcloud projects list
PROJECT_ID: melodic-casing-376708
NAME: Demo
PROJECT_NUMBER: 815361999044
harini8t@cloudshell:~$ 
```

```
You are creating an app for project [melodic-casing-376708].
WARNING: Creating an App Engine application for a project is irreversible and the region
cannot be changed. More information about regions is at
<https://cloud.google.com/appengine/docs/locations>.

Please choose the region where you want your App Engine application located:

[1] asia-east1      (supports standard and flexible)
[2] asia-east2      (supports standard and flexible and search_api)
[3] asia-northeast1 (supports standard and flexible and search_api)
[4] asia-northwest1 (supports standard and flexible and search_api)
[5] asia-northwest3 (supports standard and flexible and search_api)
[6] asia-south1      (supports standard and flexible and search_api)
[7] asia-southeast1 (supports standard and flexible)
[8] asia-southwest2 (supports standard and flexible and search_api)
[9] australia-southeast1 (supports standard and flexible and search_api)
[10] europe-central2 (supports standard and flexible)
[11] europe-west     (supports standard and flexible and search_api)
[12] europe-west2    (supports standard and flexible and search_api)
[13] europe-west3    (supports standard and flexible and search_api)
[14] europe-west6    (supports standard and flexible and search_api)
[15] northamerica-northeast1 (supports standard and flexible and search_api)
[16] southamerica-east1 (supports standard and flexible and search_api)
[17] us-central       (supports standard and flexible and search_api)
[18] us-east1         (supports standard and flexible and search_api)
[19] us-east4         (supports standard and flexible and search_api)
[20] us-west1         (supports standard and flexible)
[21] us-west2         (supports standard and flexible and search_api)
[22] us-west3         (supports standard and flexible and search_api)
[23] us-west4         (supports standard and flexible and search_api)
[24] cancel

Please enter your numeric choice: 6

Creating App Engine application in project [melodic-casing-376708] and region [asia-south1]...done.
Success! The app is now created. Please use 'gcloud app deploy' to deploy your first app.
harinist@cloudshell:~$ 
```

Welcome to App Engine

Build scalable apps in any language on Google's infrastructure

- ✓ Your App Engine application has been created

Let us help you deploy to your application by pointing you towards the relevant resources based on your programming language.

GET STARTED

RESULT

Thus the creation of hello world app and web app in using google app engine was implemented successfully

Ex:04	
Date:	

USE GAE LAUNCHER TO LAUNCH WEB APPLICATIONS

AIM

To use GAE launcher to launch web applications

PROCEDURE

- 1.Download th Google cloud SDK shell and python should be installed.
- 2.Create two files named main.py and app.yaml

Main.py:

```
app.yaml
runtime: python27
api_version: 1
threadsafe: false
handlers:
- url: /
  script: main.py
# manual_scaling:
#   instances: 10
automatic_scaling:
# Default 0 | 0 to 1000
min_instances: 0
# Default 0 | 0 to 2147483647
max_instances: 100
# Default: automatic | 1 to 1000
max_idle_instances: automatic
# Default automatic | 0 to 1000
min_idle_instances: 0
# Default: 0.6 | 0.5 to 0.95
target_cpu_utilization: 0.6
# Default: 0.6 | 0.5 to 0.95
```

```

target_throughput_utilization: 0.6
# Default 10, max 80
max_concurrent_requests: 10
# Default automatic | Range: [0.01, 15.0]. | Match expression:
'^(?:^(\d+((.\d{1,3})?s|ms)|automatic)$)$'
max_pending_latency: 1s
# Default automatic | Range: [0.01, 15.0]. | Match expression:
'^(?:^(\d+((.\d{1,3})?s|ms)|automatic)$)$'
min_pending_latency: automatic

```

3. Open Google SDK shell and run as administrator and copy and paste path of your file.

Google-cloud\bin\sdk\dev-appserver.py" path"

4. If generate the link and run the following link in web browser

http://localhost:8080

http://localhost:8000

It will create the instances

OUTPUT

```

Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.
---
C:\Program Files (x86)\Google\Cloud SDK>google-cloud-sdk\bin\dev_appserver.py "C:\Users\HariVishal\Desktop\cloudcompute"

Updates are available for some Google Cloud CLI components. To install them,
please run:
$ gcloud components update

INFO    2023-03-02 20:56:27,808 devappserver2.py:240] Using Cloud Datastore Emulator.
We are gradually rolling out the emulator as the default datastore implementation of dev_appserver.
If broken, you can temporarily disable it by --support_datastore_emulator=False
Read the documentation: https://cloud.google.com/appengine/docs/standard/python/tools/migrate-cloud-datastore-emulator
Help us validate that the feature is ready by taking this survey: https://goo.gl/forms/UArIcs8K9CUSCM733
Report issues at: https://issuetracker.google.com/issues/new?component=187272

INFO    2023-03-02 20:56:27,835 devappserver2.py:317] Skipping SDK update check.
INFO    2023-03-02 20:56:28,506 datastore_emulator.py:156] Starting Cloud Datastore emulator at: http://localhost:56177INFO
ded after 2.784000 seconds
INFO    2023-03-02 20:56:31,292 <string>:384] Starting API server at: http://localhost:56182
INFO    2023-03-02 20:56:31,318 <string>:374] Starting gRPC API server at: http://localhost:56183
INFO    2023-03-02 20:56:31,558 dispatcher.py:280] Starting module "default" running at: http://localhost:8080
INFO    2023-03-02 20:56:31,575 admin_server.py:70] Starting admin server at: http://localhost:8000
INFO    2023-03-02 20:56:37,526 instance.py:294] Instance PID: 23316
INFO    2023-03-02 20:56:37,526 module.py:1943] New instance for module "default" serving on:
http://localhost:8080

```

The screenshot shows the Google App Engine Instances dashboard at localhost:8000/instances. The left sidebar has a red 'Instances' tab selected. The main table lists 10 instances in the 'default' group. The columns are Latency (ms), QPS, Total Requests, and Runtime (python27). The first instance (id 0) has a latency of 10043.5 ms, 0.03 QPS, and 3 total requests.

	Latency (ms)	QPS	Total Requests	Runtime
0	10043.5	0.03	3	python27
1	0.0	0.00	1	
2	0.0	0.00	1	
3	0.0	0.00	1	
4	0.0	0.00	1	
5	0.0	0.00	1	
6	0.0	0.00	1	
7	0.0	0.00	1	
8	0.0	0.00	1	
9	0.0	0.00	1	

The screenshot shows a web browser window at localhost:8080. The page content is simply the text "hello world".

RESULT

Thus to launch web applications using GAE launcher was launched successfully.

Ex:05	CREATE EC2-AWS INSTANCE AND MIGRATION
Date:	

AIM

To create ec2-aws instance and migration

PROCEDURE

1. Choose the Services menu, locate the Compute services, and select EC2.
2. Choose the Launch instance button and provide details like Name the instance, Specify an Instance type:.
3. Select the key pair to associate with the instance:
 - a. From the Key pair name menu, select vockey.
4. Next to Network settings, choose Edit.
5. Keep the default VPC and subnet settings. Also keep the Auto-assign public IP setting set to Enable.
6. Under Firewall (security groups), keep the default Create security group option chosen.
7. Configure a new security group:
 - Keep the default selection Create a new security group.
 - Security group name: Clear the text and enter Web Server
 - Description: Clear the text and enter Security group for my web server
 - Choose Remove to remove the default SSH inbound rule.
8. In the Configure storage section, keep the default settings.
9. Configure a script to run on the instance when it launches:
 - Expand the Advanced details panel.
 - Scroll to the bottom of the page and then copy and paste the code shown below into the User data box:

```
#!/bin/bash
```

```
yum update -y
```

```
yum -y install httpd  
systemctl enable httpd  
systemctl start httpd  
echo '<html><h1>Hello World!</h1></html>' > /var/www/html/index.html
```

11. Select the Web Server 1 instance, and review the information in the Details tab that displays in the lower pane
12. Open a new tab in a web browser, copy and paste the public IP address
13. In the left navigation pane, under Network & Security, choose Security Groups.
14. Select the Web Server security group, which you created when launching your EC2 instance.
15. In the lower pane, choose the Inbound rules tab.
16. Choose Edit inbound rules, and then choose Add rule.
17. Configure the following:
 - Type: HTTP
 - Source: Anywhere-IPv4
 - Choose Save rules
 - The new inbound HTTP rule creates an entry for IPv4 IP (0.0.0.0/0) and IPv6 IP addresses (::/0).
18. Return to the tab that you used to try to connect to the web server and Refresh the page. The page should display the message Hello World!

OUTPUT

AWS Services Search [Alt+S] N. Virginia vocabs/user2450874=Harini_T @ 5474-0264-7621 ▾

New EC2 Experience Tell us what you think X

EC2 Dashboard EC2 Global View Events Tags Limits Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Scheduled Instances Capacity Reservations Images AMIs AMI Catalog

Instances (1/1) Info Find instance by attribute or tag (case-sensitive)

Name Instance ID Instance state Instance type Status check Alarm status Availability Zone Public IPv4 DNS

Web Server 1 i-0e52da286ea8fcfa5f Running Initializing No alarms + us-east-1e ec2-54-175-136-113

Instance: i-0e52da286ea8fcfa5f (Web Server 1)

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

Instance ID i-0e52da286ea8fcfa5f (Web Server 1)	Public IPv4 address 54.175.136.113 open address	Private IPv4 addresses 172.31.56.8
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-175-136-113.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-56-8.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-56-8.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	

AWS Services Search [Alt+S] N. Virginia vocabs/user2450874=Harini_T @ 5474-0264-7621 ▾

New EC2 Experience Tell us what you think X

EC2 Dashboard EC2 Global View Events Tags Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Scheduled Instances Capacity Reservations Images AMIs AMI Catalog Elastic Block Store Volumes Snapshots

Inbound security group rules successfully modified on security group sg-0eb28ac61544586ce | Web Server ▾ Details

Security Groups (1/2) Info Actions Export security groups to CSV Create security group

Name Security group ID Security group name VPC ID Description Owner Inbound rules count

- sg-0eb28ac61544586ce Web Server vpc-0f20ba353e533e526 Security group for my ... 547402647621 1 Permission entry

- sg-0d770be322780b015 default vpc-0f20ba353e533e526 default VPC security gr... 547402647621 1 Permission entry

sg-0eb28ac61544586ce - Web Server

Details Inbound rules Outbound rules Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

Inbound rules (1/1) Manage tags Edit inbound rules

Name Security group rule... IP version Type Protocol Port range Source

- sgr-0c37d25ea454862... IPv4 HTTP TCP 80 0.0.0/0

← → C

⚠ Not secure | 54.175.136.113

Hello World!

RESULT

Thus the ec2-aws instance and migration was created successfully

Ex:06	CREATING AN S3 BUCKET
Date:	

AIM

To create an S3 bucket

PROCEDURE

1. Choose the Services menu, locate the Storage services, and select S3.
2. Select Create bucket and enter a unique name.
3. Uncheck the Block all public access box because you want to be able to test if the website is working.
4. Below the warning, check the box next to I acknowledge that and select Create bucket.
5. Choose the link for bucket name, and then select the Permissions tab.
6. In the Bucket policy section, choose Edit.
7. To grant public read access for your website, write the code

```

"Version":"2012-10-17",
"Statement":[
{
    "Sid":"PublicReadGetObject",
    "Effect":"Allow",
    "Principal":"*",
    "Action":[
        "s3:GetObject"
    ],
    "Resource":[
        "arn:aws:s3:::example-bucket/*"
    ]
}
]
}

```

8. Replace example-bucket with the name of bucket.
9. Upload an HTML document to new bucket.

10. Choose the Objects tab and Upload your index.html file to your bucket.
11. Expand the Properties section.
 - a. Ensure that the Standard storage class is selected.
12. At the bottom of the page, choose Upload.
13. Select the Properties tab, and scroll down to the Static website hosting section.
14. Choose Edit and Select Enable.
15. In the Index document text box, enter index.html
16. Select Save changes.
17. Scroll down to the Static website hosting and copy the Bucket website endpoint URL
18. Open a new tab in your web browser, paste the URL and press Enter.

The Hello World webpage should display.

OUTPUT

The screenshot shows the 'Edit bucket policy' interface in the AWS S3 console. The left sidebar shows 'Amazon S3' and various navigation options like 'Buckets', 'Storage Lens', and 'Feature spotlight'. The main area is titled 'Edit bucket policy' and contains a 'Bucket policy' section. The policy JSON is as follows:

```

1 ~ {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "PublicReadGetObject",
6       "Effect": "Allow",
7       "Principal": "*",
8       "Action": [
9         "s3:GetObject"
10      ],
11      "Resource": [
12        "arn:aws:s3:::harinit/*"
13      ]
14    }
15  ]
16 }

```

To the right of the policy editor, there's a sidebar with sections for 'Edit statement', 'Remove', 'Add actions', 'Choose a service' (with a 'Filter services' dropdown), 'Included' (listing 'S3'), 'Available' (listing 'AMP', 'API Gateway', 'API Gateway V2', and 'ASC'), and a scrollable list of other services.

AWS Services Search [Alt+S] Global v vocabs/user2450874=Harini_T @ 2045-2703-1509

Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

- Dashboards
- AWS Organizations settings

Feature spotlight 3

AWS Marketplace for S3

Amazon S3 > Buckets > harinit > Edit static website hosting

Edit static website hosting Info

Static website hosting
Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting
 Enable
 Disable

Hosting type
 Host a static website
Use the bucket endpoint as the web address. [Learn more](#)
 Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#)

For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

Index document
Specify the home or default page of the website.
index.html

AWS Services Search [Alt+S] Global v vocabs/user2450874=Harini_T @ 2045-2703-1509

Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

- Dashboards
- AWS Organizations settings

Feature spotlight 3

AWS Marketplace for S3

Successfully edited static website hosting.

Amazon S3 > Buckets > harinit

harinit Info

Publicly accessible

Properties Permissions Metrics Management Access Points

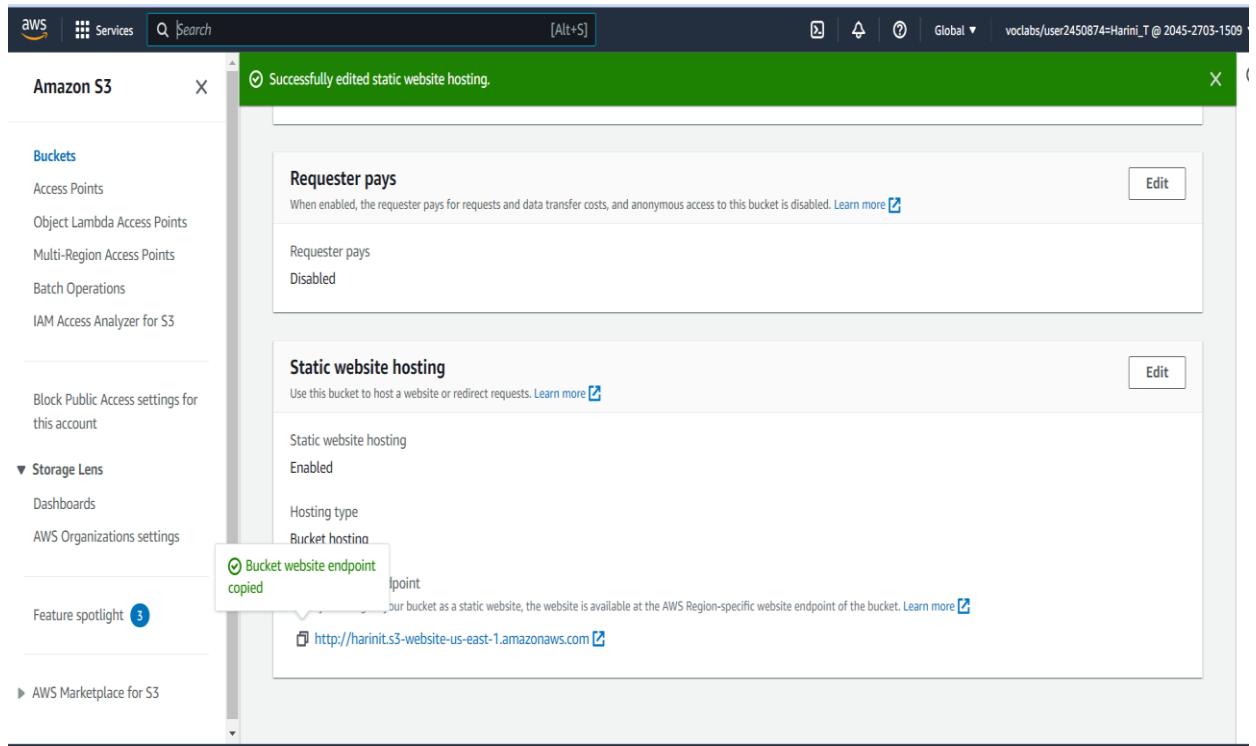
Bucket overview

AWS Region US East (N. Virginia) us-east-1	Amazon Resource Name (ARN) arn:aws:s3:::harinit	Creation date March 21, 2023, 19:26:29 (UTC+05:30)
---	--	---

Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning
Disabled

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



RESULT

Thus the S3 bucket was created successfully

Ex:07	PERFORM AWS LOAD BALANCING AND AUTO SCALING
Date:	

AIM

To perform AWS load balancing and auto scaling

PROCEDURE

1. In Services, choose EC2 to open the EC2 console and ensure you are on the dashboard page. select Launch instance
 - i. For Name enter Web Server 1
 - ii. Choose Key pair name as vockey .
2. In the Network settings section, choose Edit.
 - a. choose the existing subnet in Availability Zone us-east-1a.
 - b. For Security group name - Web Server security group
 - c. For Description - Security group for my web server
 - d. In the Inbound security groups rules section, Select Remove to remove the default rule.
 - e. Choose Add security group rule
 - i. Type : HTTP
 - ii. Source type : Anywhere
3. Expand Advanced Details and write the code in the user field
 - i. #!/bin/bash
 - ii. yum update -y
 - iii. yum -y install httpd
 - iv. systemctl enable httpd
 - v. systemctl start httpd
 - vi. echo '<html><h1>Hello World! This is server 1.</h1></html>' > /var/www/html/index.html
 - Updates the server
 - Installs an Apache web server (httpd)
 - Configures the web server to automatically start on boot
 - Starts the web server
 - Creates a simple webpage
4. Choose Launch instance and View all instances.
5. Select the Web Server 1 instance that you created
6. In the Details tab copy the Public IPv4 address of your instance, then open a new tab in your web browser and paste in and load the address.

- a. It should display the web server page with the message *Hello World! This is server 1.**
- 7. Return to the EC2 Management Console browser tab.
- 8. Select the Web Server 1 instance.
- 9. From the Actions menu, choose Images and templates, then choose Launch more like this and A Launch an instance page opens.
- 10. In the Name and tags pane, change the Name to **Web Server 2**.
- 11. Choose Key pair name as **vockey** .
- 12. choose the existing subnet in Availability Zone us-east-1b.
- 13. Expand the Advanced Details , and write the code in the User data field.

```
#!/bin/bash
yum update -y
yum -y install httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h1>Hello World! This is server 2.</h1></html>' >
/var/www/html/index.html
```

- 14. Choose Launch instance .and Choose View all instances.
- 15. Select the Web Server 2 instance.
- 16. In the Details tab copy the Public IPv4 address of your instance, then open a new tab in your web browser and paste in and load the address.
- 17. It will display the web server page with the message Hello World! This is server 2..
- 18. choose Load Balancers. And Select Create Load Balancer.
- 19. Under Application Load Balancer, choose Create.
- 20. In the Basic Configuration panel:
 - a. For Name, enter **myloadbalancer**.
- 21. In the Network mapping panel:
 - a. Under Mappings, select the Availability Zones that you created the two instances in.
For example, us-east-1a and us-east-1b.
- 22. In the Security Groups panel:
 - a. Choose Web Server security group from the drop down menu.
 - b. After you close the drop down menu, choose the X next to the default security group to remove it.
- 23. In the Listeners and routing panel:
 - a. Choose Create target group.
- 24. In the Basic Configuration panel:
 - a. Keep the target type set to Instances.
 - b. For Target group name enter **myalbTG**

25. In the Health checks panel:
 - a. For Health check path, enter `index.html` after the forward slash (/)
 - b. The path should look like the following: `/index.html`
26. Choose Next.
27. In the Register targets page, in the Available instances panel, check the boxes next to the Web Server 1 and Web Server 2 instances that you created
28. Choose Include as pending below.
 - a. verify that both instances now appear in the Targets list below.
29. Choose Create target group.
30. Return to the Load Balancers console tab in the browser.
31. In the Listeners and routing section, under Listener choose the refresh icon.
32. choose the myalbTG target group you created.
33. Choose Create load balancer.
 - a. When the load balancer is created, a Successfully created load balancer message displays.
34. Choose View load balancer.ensure that the State of the load balancer that you just created changes to Active.
35. Select the load balancer that you just created, and expand the Details section.
36. copy the DNS name value .
37. Open a new tab in your web browser, paste the DNS name that you just copied, and press Enter.
 - a. If your load balancer is working, the Hello World! message displays. Notice whether the message says This is server 1 or This is server 2.
38. Refresh the browser tab a few times.
 - a. Notice when the message changes between This is server 1 and This is server 2. When the message changes, it means that the load balancer has directed you to the web server on the other EC2 instance that you created

OUTPUT

The screenshot shows the AWS Lambda console interface. On the left, there's a sidebar with options like 'Services', 'CloudShell', 'Feedback', and 'Language'. The main area has tabs for 'Functions' and 'Events'. A search bar at the top right says 'Search [Alt+S]'. The top navigation bar includes the AWS logo, 'Services', a search bar, and account information 'N. Virginia' and 'voclabs/user2450874=Harini_T @ 2125-8997-3968'. The main content area is titled 'Summary' and shows a single instance. It includes fields for 'Number of instances' (set to 1), 'Software Image (AMI)' (Amazon Linux 2023 AMI 2023.0.2...), 'Virtual server type (instance type)' (t2.micro), and 'Storage (volumes)' (1 volume(s) - 8 GiB). A tooltip for 'Free tier' indicates it covers 750 hours of t2.micro or t3.micro usage in specific regions. At the bottom are buttons for 'Cancel', 'Launch instance', and 'Review commands'.

The screenshot shows the AWS EC2 Instances page. The left sidebar includes 'New EC2 Experience' (with a feedback link), 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', and 'Instances' (selected). Under 'Instances', there are links for 'Instance Types', 'Launch Templates', 'Spot Requests', and 'Savings Plans'. The main content area is titled 'Instances (1) Info' and shows a table with one row:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Web Server 1	i-059a7fb1b397c14b	Pending	t2.micro	-	No alarms	us-east-1a	ec2-3-80-100-14

A message 'Select an instance' is displayed below the table. At the bottom, a browser window shows the IP address 3.80.100.140 and the text 'Hello World! This is server 1.'.

A browser window with the URL 3.80.100.140. The page content is 'Hello World! This is server 1.'

A browser window with the URL 34.229.48.137. The page content is 'Hello World! This is server 2.'

aws Services Search [Alt+S]

Summary

Number of instances: **1**

Software Image (AMI)
Amazon Linux 2023 AMI 2023.0.2... [read more](#)
ami-06e46074ae430fba6

Virtual server type (instance type)
t2.micro

Firewall (security group)
Web Server security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and

Launch instance

User data has already been base64 encoded

aws Services Search [Alt+S]

Instances (2) Info

Find instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Web Server 2	i-02e1cb15001d47392	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-34-229-48-1
Web Server 1	i-059a7fb1b3979c14b	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-3-80-100-14

Actions ▾ **Launch instances** ▾

VPC Info

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

- vpc-0db6d59e90722a916
IPv4: 172.31.0.0/16

Mappings [Info](#)

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

us-east-1a (use1-az4)
Subnet: subnet-03d3914e407e3d0be

us-east-1b (use1-az6)
Subnet: subnet-0a937d50fd832285c

Screenshot of the AWS CloudFront configuration page for creating a new distribution.

gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol: HTTP

Health check path: /index.htm

Attributes
Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

Screenshot of the AWS CloudFront Register targets step.

Step 1: Specify group details

Step 2: Register targets

Available instances (2)

Instance ID	Name	State	Security groups	Zone	Subnet ID
i-02e1cb15001d47392	Web Server 2	running	Web Server security group	us-east-1b	subnet-0a937d50fd832283c
i-059a7fb1b397c14b	Web Server 1	running	Web Server security group	us-east-1a	subnet-03d5914e407e5d0be

Screenshot of the AWS EC2 Target groups page.

Target groups (1)

Name	ARN	Port	Protocol	Target type	Load balancer
myalbTG	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/myalbTG/54321	80	HTTP	Instance	None associated

Screenshot of the AWS CloudFront Listener configuration page.

Listeners and routing

Listener HTTP:80

Protocol	Port	Default action
HTTP	80	Forward to myalbTG Target type: Instance, IPv4

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag
You can add up to 50 more tags.

Add listener

The screenshot shows the AWS EC2 Load Balancers console. The left sidebar has a 'New EC2 Experience' section with a 'Tell us what you think' link. Under 'Instances', there are links for 'Instances', 'Instance Types', and 'Launch Templates'. The main area displays a table titled 'Load balancers (1)'. The table has columns for Name, DNS name, State, VPC ID, Availability Zones, Type, and Date created. A single row is shown for 'myloadbalancer', which has a DNS name of 'myloadbalancer-1547683...', a state of 'Provisioning', a VPC ID of 'vpc-04b6d59e90722a916', two availability zones, an application type, and a creation date of April 1 (UTC+0). A search bar at the top of the table says 'search: myloadbalancer'.

Not secure | myloadbalancer-412311204.us-east-1.elb.amazonaws.com

Hello World! This is server 1.

Not secure | myloadbalancer-412311204.us-east-1.elb.amazonaws.com

Hello World! This is server 2.

RESULT

Thus the AWS load balancing and auto scaling was performed successfully.

Ex:08	CREATE EC2-AWS WEB APPLICATION USING BEANSTALK
Date:	

AIM

To create ec2-aws web application using beanstalk

PROGRAM

1. Choose the Services menu, locate the Compute services, and choose Elastic Beanstalk.
2. Choose Create Application.
3. For Application name, enter a name for your application; for example, MyLabApp
4. For Platform, select PHP.
5. For Application code, select Sample application.
6. Choose Create Application
7. Open a new browser tab or window. Navigate to the [AWS Tutorials and Samples web page](#).
8. On the page, in the second list of downloads, find PHP – php.zip. Download the sample PHP application to your computer.
9. Choose Upload and deploy.
10. Choose Choose file, navigate to and select the *php.zip* file that you downloaded, and choose Open.
11. Choose Deploy.
12. To see your PHP website, in the left navigation pane, choose Go to environment.
13. Choose the Services menu, locate the Compute services, and choose EC2.
14. In the left navigation pane, under Network & Security, choose Key Pairs.
15. Choose Create key pair.
16. For Name, enter **CFLearner**
17. Choose Create key pair.
18. When the download window opens, choose Cancel. Choose the Services menu, locate the Management & Governance services, and choose CloudFormation.
19. Choose Create stack, and then choose With new resources (standard).
20. In the Prerequisite section, choose Use a sample template.

21. In the Select a sample template section, select WordPress blog.

22. Select Next.

23. For Stack name, enter **WordPressStack**

24. In the Parameters section, configure the following:

- DBPassword: Enter **Testing1**
- DBUser: Enter **testadmin**
- KeyName: Choose the CFLearner key pair

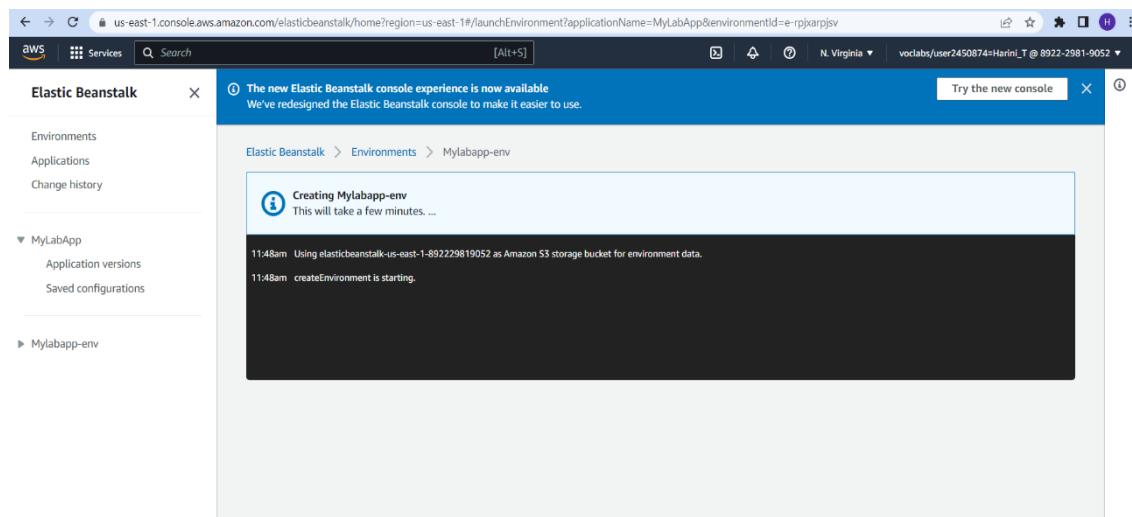
25. Choose Next, and then choose Next again.

26. Review the stack configuration, and then choose Submit.

27. We can watch the CloudFormation stack events on the Events tab as a complete

WordPress site is created.

OUTPUT



us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create

CloudFormation > Stacks > Create stack

Step 1 Create stack

Step 2 Specify stack details

Step 3 Configure stack options

Step 4 Review

Create stack

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Template is ready Use a sample template Create template in Designer

Select a sample template

View more sample templates

Sample templates
This collection of sample templates will help you get started with AWS CloudFormation and quickly build your own templates

S3 URL: https://cloudformation-templates-us-east-1.s3.us-east-1.amazonaws.com/WordPress_Multi_AZ.template

View in Designer

Cancel Next

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateKeyPair:

CloudFormation > Stacks > Create stack

Create key pair Info

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type Info
 RSA ED25519

Private key file format
 .pem For use with OpenSSH .pk For use with PUTTY

Tags - optional
No tags associated with the resource.

You can add up to 50 more tags.

Cancel

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS CloudFormation console showing the WordPressStack stack. The left sidebar shows a list of stacks, and the main panel displays the Events tab for the WordPressStack, listing six events related to the creation of resources.

Timestamp	Logical ID	Status	Status reason
2023-04-11 11:58:23 UTC+0530	ALBTargetGroup	CREATE_COMPLETE	-
2023-04-11 11:58:07 UTC+0530	ALBTargetGroup	CREATE_IN_PROGRESS	Resource creation initiated
2023-04-11 11:58:06 UTC+0530	ApplicationLoadBalancer	CREATE_IN_PROGRESS	Resource creation initiated
2023-04-11 11:58:05 UTC+0530	ALBTargetGroup	CREATE_IN_PROGRESS	-
2023-04-11 11:58:05 UTC+0530	ApplicationLoadBalancer	CREATE_IN_PROGRESS	-
2023-04-11 11:57:59 UTC+0530	WordPressStack	CREATE_IN_PROGRESS	User Initiated

Screenshot of the AWS Elastic Beanstalk console showing the upload and deployment process for the MyLabApp environment. A modal window titled "Upload and deploy" is open, prompting the user to choose a file to upload. The application version is set to "php.zip". Deployment preferences indicate "All at once" policy with 1 instance.

Screenshot of the AWS CloudFormation "Create New Stack" wizard, step 2: Set the template parameters. The "t2.small" instance type is selected. Other fields include:

- KeyName:** CFLearnert
- MultiAZDatabase:** false
- SSHLocation:** 0.0.0.0/0
- Subnets:** subnet-020736e246b95f094, subnet-0f34d3fa4c62586af
- VpcId:** vpc-0a9ed784a77c2bdb8
- WebServerCapacity:** 1

Screenshot of the AWS Elastic Beanstalk console showing the environment "Mylabapp-env". The status is "Ok" with a green checkmark icon. The running version is "Sample Application". The platform is "PHP 8.1 running on 64bit Amazon Linux 2/3.5.6". Recent events show a successful launch on 2023-04-11.

Screenshot of the AWS Elastic Beanstalk configuration page for the application "MyLabApp". It shows the application information and tags section, where a tag "MyLabApp" is added. The platform section shows PHP 8.1 running on 64bit Amazon Linux 2.

Screenshot of the AWS Elastic Beanstalk environment URL (<https://mylabapp-env.eba-ztlfpkdez.us-east-1.elasticbeanstalk.com>). The page displays a large "Congratulations!" message, stating that the PHP application is now running on its own dedicated environment in the AWS Cloud. It also mentions the PHP version (8.1.17) and the Platform (Elastic Beanstalk PHP). On the right side, there are sections for "What's Next?" and "AWS SDK for PHP".

RESULT

Thus the EC2-aws web application using beanstalk was created successfully.

Ex:09	
Date:	

**INSTALL HADOOP SINGLE NODE CLUSTER
AND RUN SIMPLE APPLICATIONS LIKE
WORDCOUNT.**

AIM

To install hadoop single node cluster and run simple applications like wordcount.

PROCEDURE

Program

Mapper.java

```
package com.javatpoint;

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.MapReduceBase;

import org.apache.hadoop.mapred.Mapper;

import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reporter;

public class WC_Mapper extends MapReduceBase implements Mapper<LongWritable,Te xt,Text,Int Writable>{

    private final static IntWritable one = new IntWritable(1);

    private Text word = new Text();

    public void map(LongWritable key, Text value,OutputCollector<Text,IntWritable> o utput,
    Reporter reporter) throws IOException{
```

```
String line = value.toString();

 StringTokenizer tokenizer = new StringTokenizer(line);

 while (tokenizer.hasMoreTokens()){

    word.set(tokenizer.nextToken());

    output.collect(word, one);

 }

}

}
```

Reducer.java

```
package com.javatpoint;

import java.io.IOException;

import java.util.Iterator;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.MapReduceBase;

import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reducer;

import org.apache.hadoop.mapred.Reporter;

public class WC_Reducer extends MapReduceBase implements Reducer<Text,Int

Writable,Text,IntWritable> {

    public void reduce(Text key, Iterator<IntWritable> values,OutputCollector<Text,I

nt

Writable>, reporter) throws IOException {

    int sum=0;

    while (values.hasNext()) {
```

```
sum+=values.next().get();

}

output.collect(key,new IntWritable(sum));

}

}
```

Runner.java

```
import java.io.IOException;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.FileInputFormat;

import org.apache.hadoop.mapred.FileOutputFormat;

import org.apache.hadoop.mapred.JobClient;

import org.apache.hadoop.mapred.JobConf;

import org.apache.hadoop.mapred.TextInputFormat;

import org.apache.hadoop.mapred.TextOutputFormat;

public class WC_Runner {

    public static void main(String[] args) throws IOException{

        JobConf conf = new JobConf(WC_Runner.class);

        conf.setJobName("WordCount");

        conf.setOutputKeyClass(Text.class);

        conf.setOutputValueClass(IntWritable.class);

        conf.setMapperClass(WC_Mapper.class);

        conf.setCombinerClass(WC_Reducer.class);

        conf.setReducerClass(WC_Reducer.class);
```

```
    conf.setInputFormat(TextInputFormat.class);
    conf.setOutputFormat(TextOutputFormat.class);
    FileInputFormat.setInputPaths(conf,new Path(args[0]));
    FileOutputFormat.setOutputPath(conf,new Path(args[1]));
    JobClient.runJob(conf);
}
}
```

STEPS:

- 1.Go to dockerplayground and create new instance.
- 2.Copy the ssh-key
- 3.Open the terminal and run the command:ssh-keygen
- 4.Paste the ssh-key in the terminal.
- 5.Run the following commands:

```
docker pull sequenceiq/hadoop-docker:latest
docker run -it -p 50070:50070 -p 8088:8088 -p 8042:8042 sequenceiq/hadoop-docker:latest
/etc/bootstrap.sh -bash
cd usr/local/hadoop/
vi input/data.txt
```

To create a new text file -cat input/data.txt and write content inside the file

```
bin/hadoop fs -mkdir /input
bin/hadoop fs -put input/data.txt /input
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.0.jar wordcount
/input/data.txt /output
```

To execute -bin/hdfs dfs -cat /output/*

OUTPUT

```
C:\Users\HariVishal>ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\HariVishal/.ssh/id_rsa):
Created directory 'C:\Users\HariVishal/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\HariVishal/.ssh/id_rsa.
Your public key has been saved in C:\Users\HariVishal/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:KdkDPplv98PJEr0Bk5Dk2C63n6BTQY9sijceBPCKMwE harivishal@LAPTOP-Q8Q0OJJC
The key's randomart image is:
+---[RSA 3072]---+
|E .. .+=+ |
| . .. =o..*o. |
| . ooooo. *o |
| o o B.o= .... |
| + . BoSoO. |
| o .o+=.. |
| o +o |
| o. o. |
| ... o |
+---[SHA256]---+
C:\Users\HariVishal>ssh ip172-18-0-21-ch15vuo1k7jg00bbsgrg@direct.labs.play-with-docker.com
The authenticity of host 'direct.labs.play-with-docker.com (40.121.45.0)' can't be established.
RSA key fingerprint is SHA256:vCbhEjmYQhtNzuuh7TmGAPfe9t+JdGnGaVquQIUphY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

```
C:\Users\HariVishal>ssh ip172-18-0-21-ch15vuo1k7jg00bbsgrg@direct.labs.play-with-docker.com
Connecting to 20.25.0.130:8022
#####
#          WARNING!!!!          #
# This is a sandbox environment. Using personal credentials      #
# is HIGHLY! discouraged. Any consequences of doing so are      #
# completely the user's responsibilites.                         #
#                                                               #
# The PWD team.                                              #
#####
```

```
/deprecated-schema-v1/
b253335dcf03: Pull complete
a3ed95caeb02: Pull complete
69623ef05416: Pull complete
8d2023764774: Pull complete
0c3c0ff61963: Pull complete
ff0696749bf6: Pull complete
72accdc282f3: Pull complete
5298ddb3b339: Pull complete
f252bbba6bda: Pull complete
3984257f0553: Pull complete
26343a20fa29: Pull complete
```

```
247c8fd8465a: Pull complete
e00909753b86: Pull complete
18f53e764edf: Pull complete
3a8c9b2d833a: Pull complete
770ef75e185a: Pull complete
0173662eb77d: Pull complete
c0a37ad8136f: Pull complete
47eeef0823a8: Pull complete
77c1cad6eb69: Pull complete
ed6629089518: Pull complete
36a49c5cc0d9: Pull complete
e6a7899cd72b: Pull complete
Digest: sha256:5a971a61840d9d32752330a891d528d35a558adf31d99c46205f1180af8e1abd
Status: Downloaded newer image for sequenceiq/hadoop-docker:latest
docker.io/sequenceiq/hadoop-docker:latest
```

```

Digest: sha256:5a971a61840d9d32752330a891d528d35a558adf31d99c46205f1180af8e1abd
Status: Downloaded newer image for sequenceiq/hadoop-docker:latest
docker.io/sequenceiq/hadoop-docker:latest
[node1] (local) root@192.168.0.13 ~
$ docker run -it -p 50070:50070 -p 8088:8088 -p 8042:8042 sequenceiq/hadoop-docker:latest /etc/bootstrap.sh -bash
/
Starting sshd: [ OK ]
Starting namenodes on [453acb709539]
453acb709539: starting namenode, logging to /usr/local/hadoop/logs/hadoop-root-namenode-453acb709539.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode-453acb709539.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-root-secondarynamenode-453acb709539.out
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn--resourcemanager-453acb709539.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-453acb709539.out
bash-4.1# cd usr/local/hadoop/
bash-4.1# vi input/data.txt
bash-4.1# cat input/data.txt
Hello Harini Hello Harini
bash-4.1# bin/hadoop fs -mkdir /input
bash-4.1# bin/hadoop fs -put input/data.txt /input
bash-4.1# bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.0.jar wordcount /input/data.txt /output
23/04/21 06:28:24 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/04/21 06:28:29 INFO InputFormat: Total input paths to process : 1
23/04/21 06:28:30 INFO mapreduce.JobSubmitter: number of splits:1
23/04/21 06:28:32 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1682072680308_0001
23/04/21 06:28:35 INFO impl.YarnClientImpl: Submitted application application_1682072680308_0001
23/04/21 06:28:35 INFO mapreduce.Job: The url to track the job: http://453acb709539:8088/proxy/application_1682072680308_0001/
23/04/21 06:28:35 INFO mapreduce.Job: Running job: job_1682072680308_0001
23/04/21 06:29:33 INFO mapreduce.Job: Job job_1682072680308_0001 running in uber mode : false
23/04/21 06:29:33 INFO mapreduce.Job: map 0% reduce 0%
23/04/21 06:30:19 INFO mapreduce.Job: map 100% reduce 0%
23/04/21 06:30:41 INFO mapreduce.Job: map 100% reduce 100%
23/04/21 06:30:44 INFO mapreduce.Job: Job job_1682072680308_0001 completed successfully
23/04/21 06:30:45 INFO mapreduce.Job: Counters: 49
File System Counters
```

Select OpenSSH SSH client

1 Hello Harini Hello Harini

```

23/04/21 06:30:45 INFO mapreduce.Job: Counters: 49
File System Counters
    FILE: Number of bytes read=31
    FILE: Number of bytes written=230311
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=130
    HDFS: Number of bytes written=17
    HDFS: Number of read operations=6
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
Job Counters
    Launched map tasks=1
    Launched reduce tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=41338
    Total time spent by all reduces in occupied slots (ms)=20055
    Total time spent by all map tasks (ms)=41338
    Total time spent by all reduce tasks (ms)=20055
    Total vcore-seconds taken by all map tasks=41338
    Total vcore-seconds taken by all reduce tasks=20055
    Total megabyte-seconds taken by all map tasks=42330112
    Total megabyte-seconds taken by all reduce tasks=20536320
Map-Reduce Framework
    Map input records=1
    Map output records=4
    Map output bytes=42
    Map output materialized bytes=31
    Input split bytes=104
    Combine input records=4
    Combine output records=2
    Reduce input groups=2
    Reduce shuffle bytes=31
    Reduce input records=2
    Reduce output records=2
    Spilled Records=4
    Shuffled Maps =1
    Failed Shuffles=0
    Merged Map outputs=1
    GC time elapsed (ms)=3687
    CPU time spent (ms)=2600
    Physical memory (bytes) snapshot=430731264
    Virtual memory (bytes) snapshot=1506664448
    Total committed heap usage (bytes)=402653184
```

```
Total committed heap usage (bytes)=4026
Shuffle Errors
    BAD_ID=0
    CONNECTION=0
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
File Input Format Counters
    Bytes Read=26
File Output Format Counters
    Bytes Written=17
bash-4.1# bin/hdfs dfs -cat /output/*
Harini 2
Hello 2
bash-4.1#
```

RESULT

Thus the installation hadoop single node cluster and run simple applications like wordcount was implemented successfully.

Ex:10	
Date:	

IMPLEMENT PAAS-MOBILE SENSOR BASED IOT APPLICATION HOSTED VIA PAAS ENVIRONMENT

AIM

To implement paas-mobile sensor based iot application hosted via paas environment

PROCEDURE

1. Download blynk app in your mobile.
2. Signup with your mail mail id.
3. Open the browser in the laptop with the same mail id used in blynk app.
4. Open <https://wokwi.com/> choose ESP32 + DHT22 Sensor template.
5. Go to library manager then click “+” icon . Search WIFI and select it. Then search blynk then select it.
6. Copy the following code and paste it in esp32-dh22.ino

```
#define BLYNK_TEMPLATE_ID "TMPL3gzb7xkjV"
#define BLYNK_TEMPLATE_NAME "PAAS"
#define BLYNK_AUTH_TOKEN "aJ_KsN6wGmbZ7s2GnBc1opWwCuKdp4zS"
#define BLYNK_PRINT Serial
#define LED 26
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <DHTesp.h>
char auth[] = BLYNK_AUTH_TOKEN ;
char ssid[] = "Wokwi-GUEST";
char pass[] = "";
const int DHT_PIN = 15;
DHTesp dht;
BlynkTimer timer;
void sendSensor()
{
    TempAndHumidity data = dht.getTempAndHumidity();
```

```
Serial.print("Temperature: ");
Serial.print(data.temperature);
Serial.println(" C ");
Serial.print("Humidity: ");
Serial.print(data.humidity);
Serial.println(" % ");
Blynk.virtualWrite(V0, data.temperature);
Blynk.virtualWrite(V1, data.humidity);
}

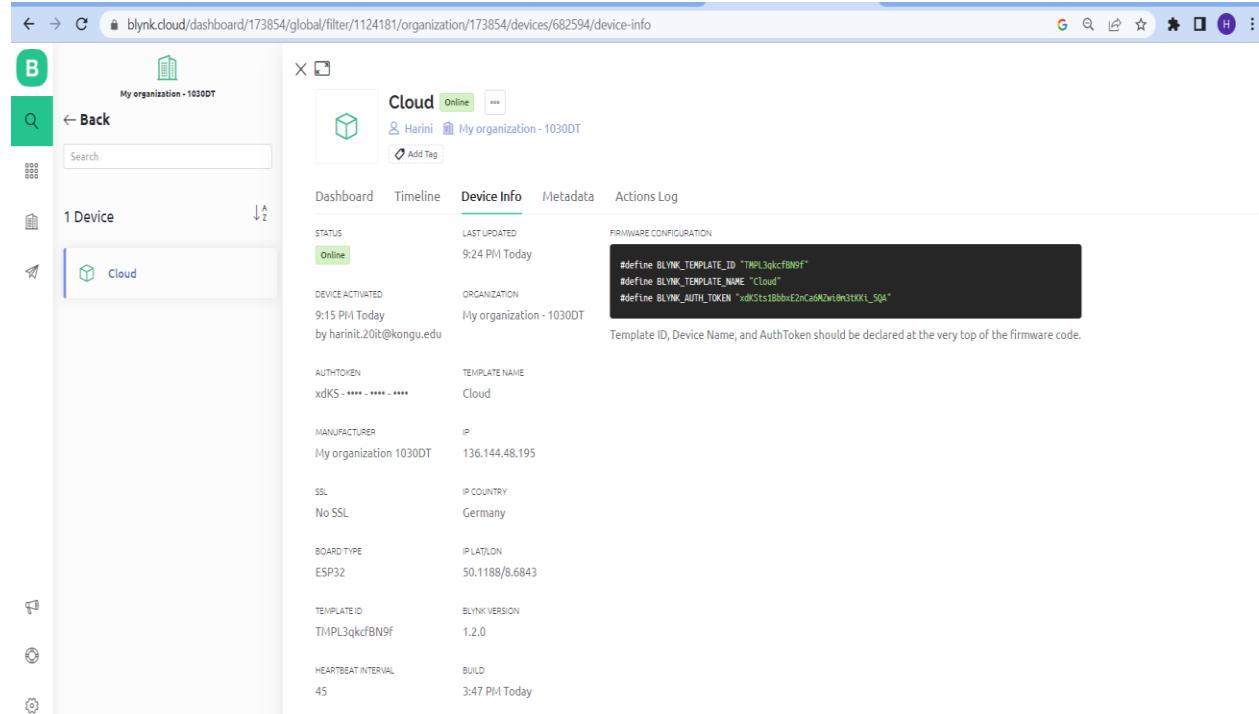
int SW_State=0;
BLYNK_WRITE (V2)
{
SW_State = param.asInt();
if (SW_State == 1)
{
digitalWrite(LED, HIGH);
Serial.println("LAMP ON");
Blynk.virtualWrite(V1, HIGH);
}
else
{
digitalWrite(LED, LOW);
Serial.println("LAMP OFF");
Blynk.virtualWrite(V1, LOW);
}
}

void setup()
{
// Debug console
Serial.begin(115200);
dht.setup(DHT_PIN, DHTesp::DHT22);
```

```
Blynk.begin(auth, ssid, pass);  
timer.setInterval(1000, sendSensor);  
pinMode(LED, OUTPUT);  
}  
  
void loop()  
{  
    Blynk.run();  
    timer.run();  
}
```

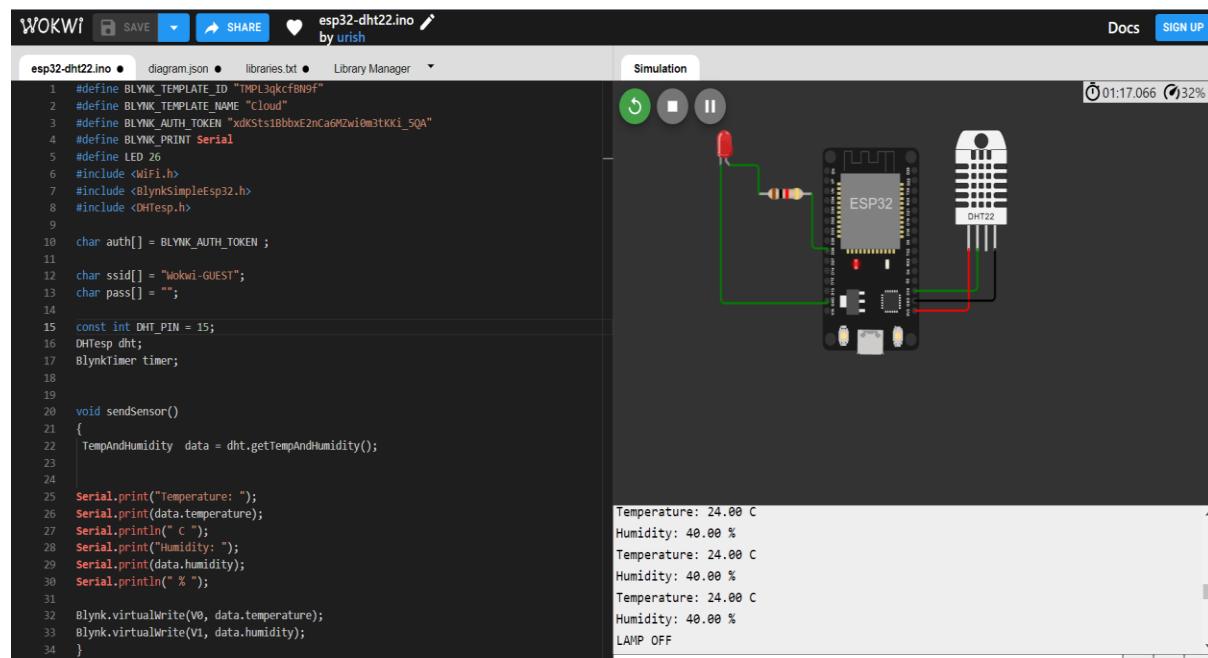
7. In the simulation section, click plus symbol then choose a resistor and a led.
8. Connect one side of the led pin to the resistor and another to ground.
9. Connect one side of the resistor to led and another to D26 pin.
10. In the blynk mobile app, create new template with the template name as sample then choose the created template.
11. Add widgets such as button,labeled value, led,gauge.
12. Click the labeled value choose datastream delete v1 and v2 and click create new,choose data type as double max value to 100, unit as celsius and click done and choose designs -> Give title and colours.
13. Click the gauge choose datastream click create new ,choose data type as double max value to 100, unit as percentage and click done.
14. Click the led choose data stream click create new , set max value to 1 and click done.
15. Click the button choose data stream as v2 click done.
16. Return to home page, click the plus symbol then choose manually from a template option then choose you template and then click create device.
17. Open blynk in your browser,choose your template and click device info -> copy the code and replace it in the first 3 lines of the existing code in wokwi workplace.
18. Start the simulation and control using your mobile app

OUTPUT



A screenshot of the Blynk Cloud interface showing device information for an ESP32 connected via WiFi. The device is online and part of the organization 'My organization - 1030DT'. The device info table includes fields like AUTH TOKEN, MANUFACTURER, BOARD TYPE, and HEARTBEAT INTERVAL. A code editor on the right shows the top of the firmware code with Blynk configuration macros.

STATUS	LAST UPDATED	FIRMWARE CONFIGURATION
Online	9:24 PM Today	#define BLYNK_TEMPLATE_ID "TMPL3qkfBN9f" #define BLYNK_TEMPLATE_NAME "Cloud" #define BLYNK_AUTH_TOKEN "xdKSts1BbbxE2nCa6Mzwiem2tKKi_5Q"
DEVICE ACTIVATED	ORGANIZATION	
9:15 PM Today	My organization - 1030DT	
by harini.201t@kongu.edu		Template ID, Device Name, and AuthToken should be declared at the very top of the firmware code.
AUTH TOKEN	TEMPLATE NAME	
xdKS-*****-*****	Cloud	
MANUFACTURER	IP	
My organization 1030DT	136.144.48.195	
SSL	IP COUNTRY	
No SSL	Germany	
BOARD TYPE	IP LAT/LON	
ESP32	50.1188/8.6843	
TEMPLATE ID	BLYNK VERSION	
TMPL3qkfBN9f	1.2.0	
HEARTBEAT INTERVAL	BUILD	
45	3:47 PM Today	



A screenshot of the Wokwi simulation environment for an ESP32-DHT22 project. The code editor contains the main sketch file 'esp32-dht22.ino' which includes Blynk library configurations and a DHT sensor reading loop. The simulation window shows the physical connections between the ESP32 board, a DHT22 sensor, and a red LED. The serial monitor on the right displays the real-time sensor data and Blynk command responses.

```
#include <WiFi.h>
#include <blynkSimpleEsp32.h>
#include <DHTesp.h>

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Wokwi-GUEST";
char pass[] = "";

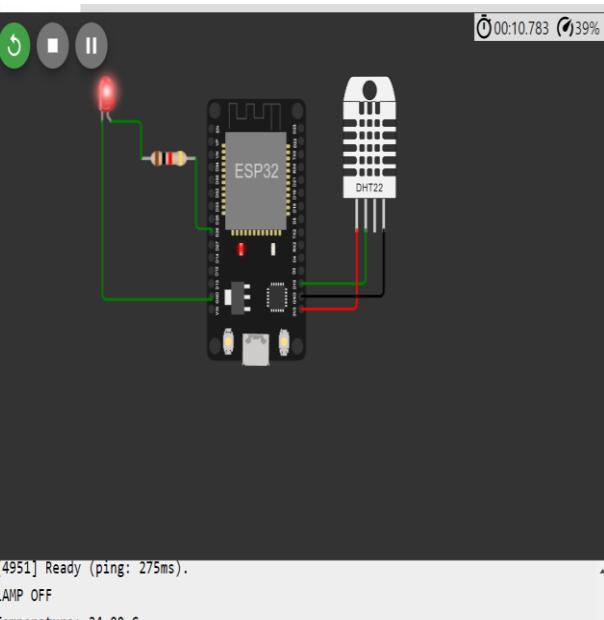
const int DHT_PIN = 15;
DHTesp dht;
BlynkTimer timer;

void sendSensor()
{
    TempAndHumidity data = dht.getTempAndHumidity();

    Serial.print("Temperature: ");
    Serial.print(data.temperature);
    Serial.println(" C");
    Serial.print("Humidity: ");
    Serial.print(data.humidity);
    Serial.println(" %");

    Blynk.virtualWrite(V0, data.temperature);
    Blynk.virtualWrite(V1, data.humidity);
}
```

Temperature: 24.00 C
Humidity: 40.00 %
Temperature: 24.00 C
Humidity: 40.00 %
Temperature: 24.00 C
Humidity: 40.00 %
LAMP OFF

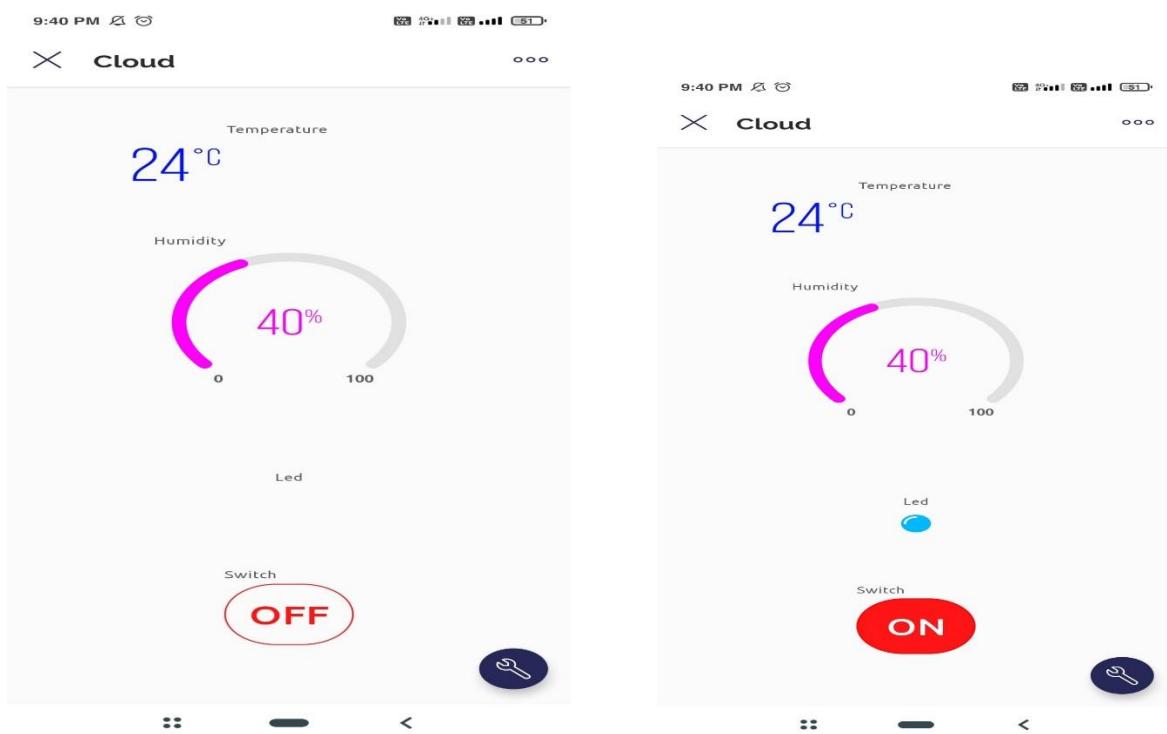


```
1 #define BLYNK_TEMPLATE_ID "TMPL3qkcfBN9f"
2 #define BLYNK_TEMPLATE_NAME "Cloud"
3 #define BLYNK_AUTH_TOKEN "xdKSts1BbbxE2nCa6M2wi0m3tKKi_5Qa"
4 #define BLYNK_PRINT Serial
5 #define LED 26
6 #include <WiFi.h>
7 #include <BlynkSimpleEsp32.h>
8 #include <DHTesp.h>
9
10 char auth[] = BLYNK_AUTH_TOKEN ;
11
12 char ssid[] = "Wokwi-GUEST";
13 char pass[] = "";
14
15 const int DHT_PIN = 15;
16 DHTesp dht;
17 BlynkTimer timer;
18
19
20 void sendSensor()
21 {
22     TempAndHumidity data = dht.getTempAndHumidity();
23
24
25     Serial.print("Temperature: ");
26     Serial.print(data.temperature);
27     Serial.println(" C ");
28     Serial.print("Humidity: ");
29     Serial.print(data.humidity);
30     Serial.println(" % ");
31
32     Blynk.virtualWrite(V0, data.temperature);
33     Blynk.virtualWrite(V1, data.humidity);
34 }
```

[4951] Ready (ping: 275ms).
LAMP OFF
Temperature: 24.00 C
Humidity: 40.00 %
Temperature: 24.00 C
Humidity: 40.00 %



```
OpenSSH SSH client
1 Hello Harini Hello Harini
~
```



RESULT

Thus the paas-mobile sensor based iot application hosted via paas environment was implemented successfully